Amendments to the Claims:

1. (Currently amended) A system for distributing and enforcing use of digital documents having usage rights associated therewith, said system comprising:

a server having at least one document stored thereon in computer readable form;

a client having a standard application program including a rendering engine capable of rendering unencrypted documents for viewing;

a communications network coupled to said client and said server;

a rights management module included in said server and configured for receiving a request for at least one of the documents from said client and delivering the at least one document and a set of rights associated with and for enforcing use of the at least one document to said client;

a connection module which is downloaded and included in said client and is ~~adapted to be~~ attached to but separate from said rendering engine for receiving the ~~list~~ set of rights associated with the at least one document, for verifying the integrity of the client by confirming a user interface module is attached to said rendering engine, and for verifying the integrity of the rendering engine;

a user interface module which is downloaded and included in said client and is ~~adapted to be~~ attached to but separate from said rendering engine for controlling access by the client to the at least one document for enforcing use of the at least one document in accordance with the set of rights associated with said at least one document.

2. (Previously presented) A system as recited in claim 1, wherein said connection module is operative to detect whether said user interface module is attached to said rendering engine and for providing the at least one of the documents to said rendering engine if said user interface module is attached to said rendering drive.

3. (Previously presented) A system as recited in claim 2, wherein said connection module is operative to unencrypt the at least one of the documents.

4. (Previously presented) A system as recited in claim 2, wherein said standard application program is a Web browser and said server includes an HTTP server.

5. (Previously presented) A system as recited in claim 4, wherein said connection module and said user interface module attach to the rendering engine of the Web browser using at least one of ActiveX controls and plug-in technology.

6. (Previously presented) A system as recited in claim 4, wherein said rights management module comprises means for pointing to a start Web page stored on said server, and means for encrypting said means for pointing and wherein said connection module comprises means for unencrypting said means for pointing and wherein said system further comprises means for generating a secure start Web page which references said connection module and said means for pointing.

7. (Previously presented) A system as recited in claim 4, wherein said connection module comprises means for generating a signature and said rights management module comprises means for validating the signature, and wherein a request to said server is honored only if the signature is present and valid.

8. (Currently amended) A method for distributing and enforcing use of digital documents having one or more usage rights associated therewith, said method comprising the steps of:

storing at least one document on a server in computer readable form;

accessing the server with a client having a standard application program including a rendering engine capable of rendering unencrypted documents;

receiving by a rights management module included in said server a request for at least one of the documents from the client;

delivering by said rights management module the at least one of the documents and a set of rights associated with and for enforcing use of the at least one of the documents to the client;

downloading a connection module and a user interface module to said client;

receiving the list of rights associated with the at least one of the documents with [[a]] said connection module included in said client and attached to but separate from the rendering engine and that verifies the integrity of the client by confirming a user interface module is attached to said rendering engine and verifies the integrity of the rendering engine;

controlling access by the client to the at least one of the documents for enforcing use of the at least one document in accordance with the set of rights associated with the at least one of the documents through [[a]] said user interface module included in said client and attached to

but separate from the rendering engine.

9. (Previously presented) A method as recited in claim 8, further comprising the step of unencrypting the at least one of the documents.

10. (Previously presented) A method as recited in claim 8, further comprising the steps of detecting whether the user interface module is attached to the rendering engine and providing the at least one document to the rendering engine if the user interface module is attached to the rendering drive.

11. (Previously presented) A method as recited in claim 10, wherein said step of detecting further comprises determining whether said rendering engine has been compromised.

12. (Previously presented) A method as recited in claim 10, wherein said standard application program is a Web browser and said server includes HTTP server software.

13. (Previously presented) A method as recited in claim 12, further comprising the steps of providing a pointer on the server to a start Web page stored on the server, encrypting the pointer, generating a secure start Web page on the server which references the pointer, providing access to the secure start Web page through the Web browser, and unencrypting the pointer on the client to provide the Web browser access to the start Web page on the server.

14. (Previously presented) A method as recited in claim 12, further comprising the steps of generating a signature with the client, transmitting the signature to the server with a request to the server, validating the signature with the server, and honoring the request only if the signature is present and valid.

15. (Previously presented) In a computer architecture including a server having documents stored thereon, a start page for accessing the documents, and a client running an application program having a rendering engine, a method of distributing and enforcing use of documents comprising the steps of:

installing a rights management module on the server;

downloading a connection module and a user interface module to said client;

attaching said [[a]] user interface module and [[a]] said connection module to the rendering engine,

wherein said user interface module and said connection module are separate from said rendering engine;

creating a secure start page on the server;

placing the documents in directory;

programming the rights management module to include a pointer to the directory;

encrypting an address to the directory;

modifying the secure interface display to reference the user interface module and the start page;

verifying the integrity of the client with the connection module by confirming the user interface module is attached to the rendering engine;

verifying the integrity of the rendering engine with the connection module;

unencrypting the address to the directory with the connection module to permit access to the start page and the documents on the server; and

enforcing use of the documents with the user interface module in accordance with a set of rights associated with the documents.

16. (Previously presented) A method as recited in claim 15, wherein the server includes HTTP server software, wherein the application program is a Web browser, wherein the secure interface display is a secure start Web page and wherein the address to the directory is in the form of a URL.

17. (Previously presented) A method as recited in claim 16, further comprising the steps of:

accessing the secure start Web page by issuing a URL to the start page;

directing the user interface module to the start page through the reference to the start page in the secure start Web page;

creating an instance of the rendering engine;

loading the start page in the instance of the rendering engine to display the start page on the client; and

directing the instance of the rendering engine, under control of the user interface module, to retrieve one or more of the documents from the server.

18. (Previously presented) A method as recited in claim 16, wherein said step of directing the instance comprises the steps of intercepting commands from the Web browser with the user interface module and redirecting the commands through the connection module on the server.

19. (Previously presented) A method as recited in claim 16, wherein said step of redirecting comprises the steps of instructing the instance to utilize a secure asynchronous protocol through the connection module.

20. (Previously presented) A method as recited in claim 16, further comprising the steps of validating, with the connection module, that the user interface module is attached to the rendering engine and permitting the client to connect to the server only if the validation step is positive.

21. (Previously presented) A system as recited in claim 1, wherein the connection module verifies the integrity of the rendering engine by verifying that the rendering engine has not been tampered with or otherwise compromised in a way that allows access to the at least one document in a way that bypasses the user interface module.

22. (Previously presented) A system as recited in claim 1, wherein the rendering engine does not have direct access to the at least one document because the rendering engine is wrapped by the user interface module to prevent the rendering engine from performing prohibited functions outside of a scope of the set of rights associated with and for enforcing the use of the at least one document.

23. (Previously presented) A method as recited in claim 8, wherein the step of verifying the integrity of the rendering engine includes verifying that the rendering engine has not been tampered with or otherwise compromised in a way that allows access to the at least one document in a way that bypasses the user interface module.

24. (Previously presented) A method as recited in claim 8, further comprising wrapping the rendering engine with the user interface module so that the rendering engine does not have

direct access to the at least one document to prevent the rendering engine from performing prohibited functions outside of a scope of the set of rights associated with and for enforcing the use of the at least one document.

25. (Previously presented) A method as recited in claim 15, wherein the step of verifying the integrity of the rendering engine includes verifying that the rendering engine has not been tampered with or otherwise compromised in a way that allows access to the documents in a way that bypasses the user interface module.

26. (Previously presented) A method as recited in claim 15, further comprising wrapping the rendering engine with the user interface module so that the rendering engine does not have direct access to the documents to prevent the rendering engine from performing prohibited functions outside of a scope of the set of rights associated with and for enforcing the documents.